Artificial Intelligence

Visual Programming for ~~Data Science~~

Working with Data at the right Level of Abstraction

Prof. Dr. Michael Berthold

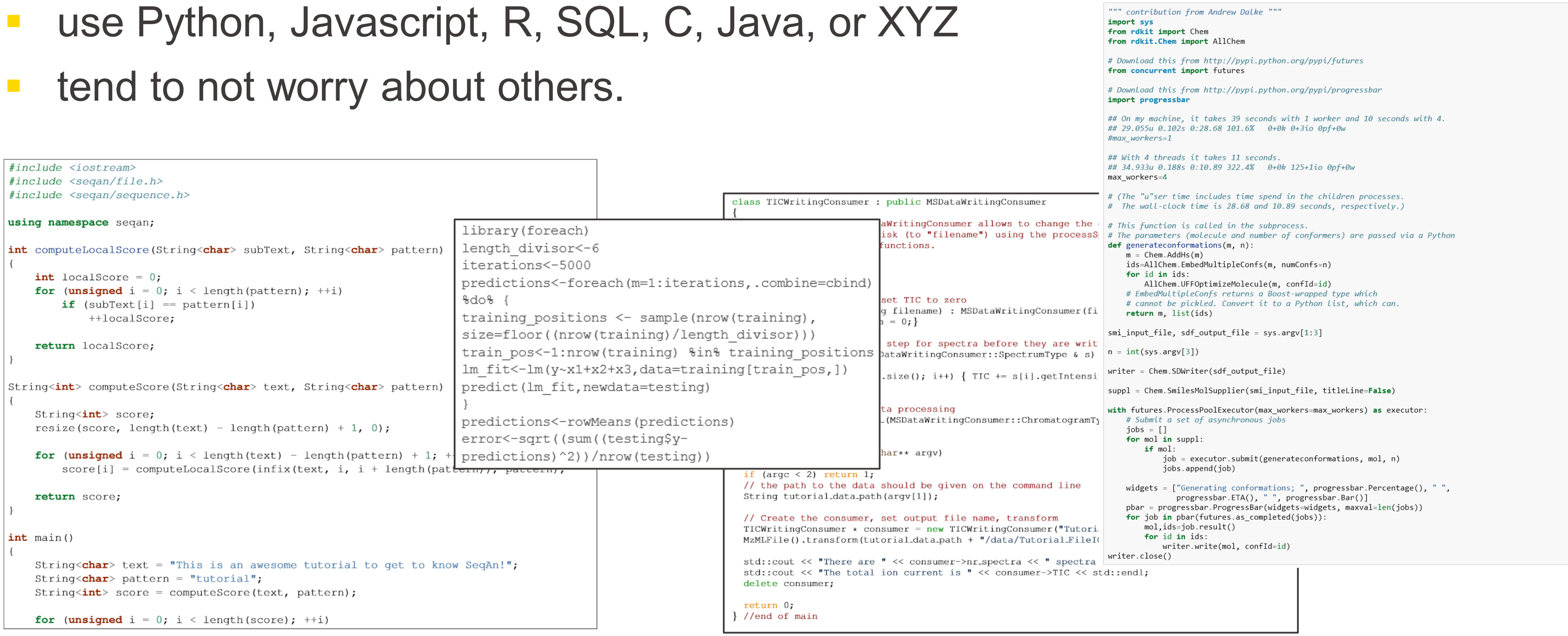
ISACA Fokus Event, Nov 10, 2022

Data Science Personas

Data Science Researchers

Researchers focus on Data Science algorithms

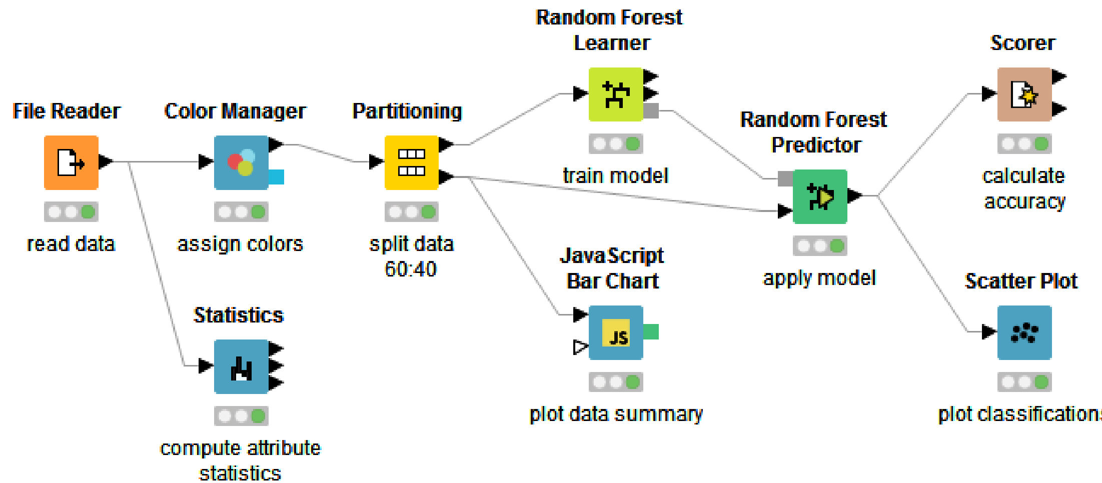
- often have CS/NumMath/Stats background
- use Python, Javascript, R, SQL, C, Java, or XYZ
- tend to not worry about others.



Data Scientists

Data Scientists

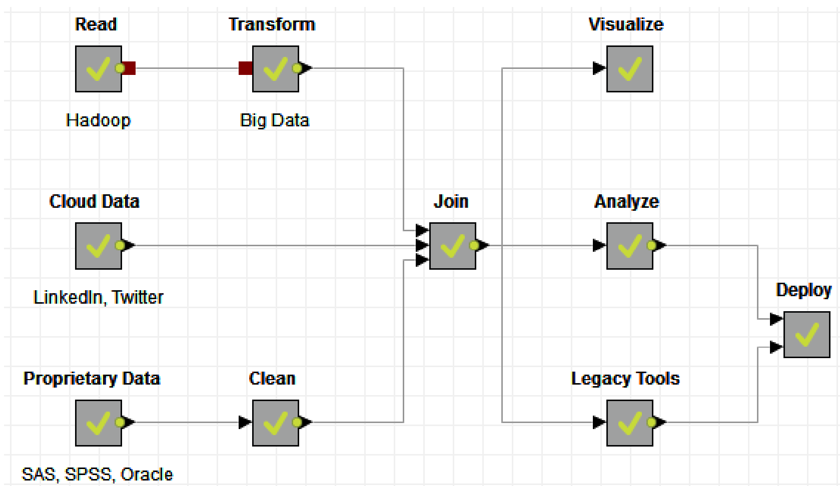
- have (also) a specific domain background
- often can program in Python, R, SQL, C, Java, or ...
- …but really focus on the data flow
- want to use and try out other stuff

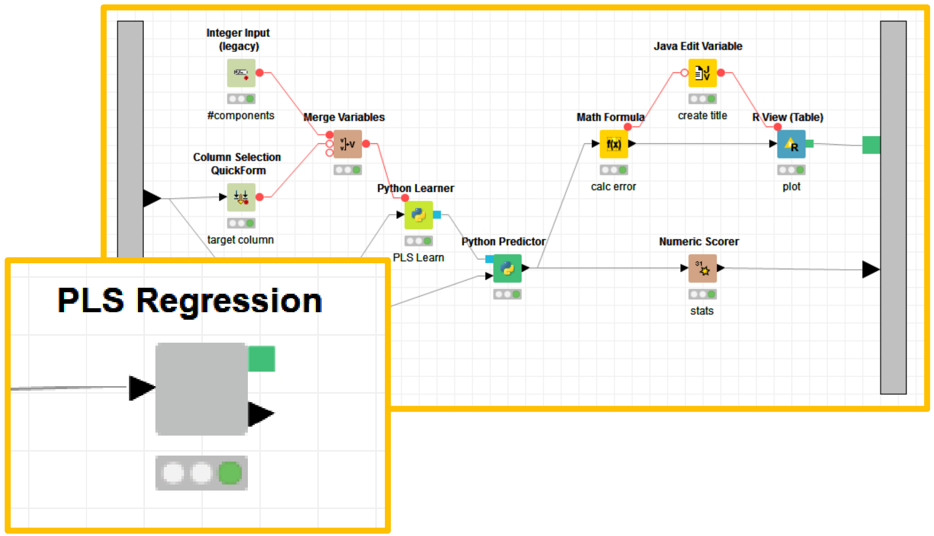


Casual Users (“Citizen Data Scientists”)

The occasional data person

- wants to start from something existing that solves a similar problem
- often comes from an Excel, BI, or other data world
- needs solid documentation and proper abstraction





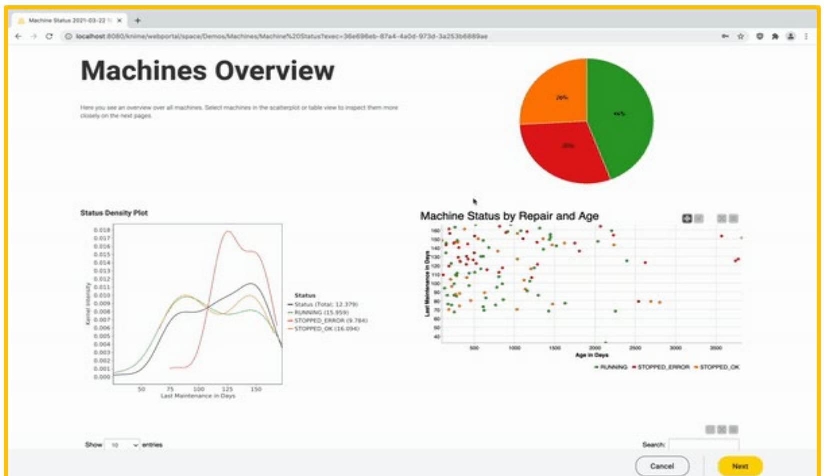
Data Science Consumers

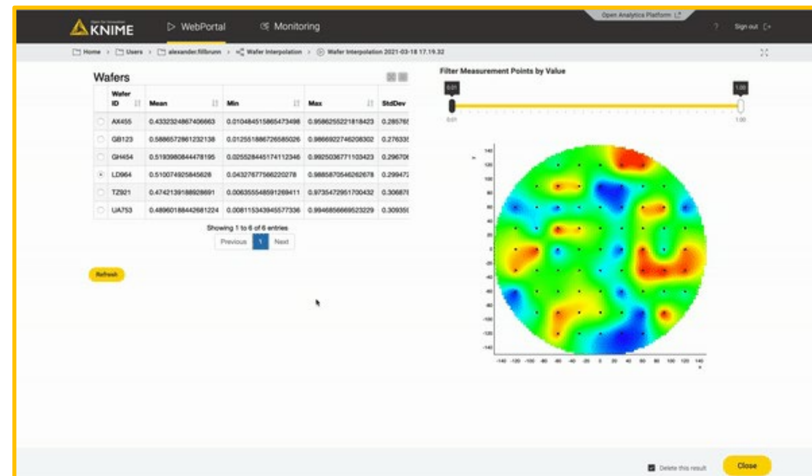
Data Science Consumers use

- their favorite frontend
  (= Data Science needs to embedded via an API)

or

- Interactive Analytics (= Data Science Applications)



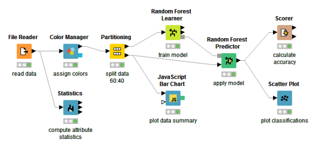


Data Science Personas

```
library(foreach)
length_divisor<-6
iterations<-5000
predictions<-foreach(m=1:iterations,.combine=cbind)
%do% {
training_positions <- sample(nrow(training),
size=floor((nrow(training)/length_divisor)))
train_pos<-1:nrow(training) %in% training_positions
lm_fit<-lm(y~x1+x2+x3,data=training[train_pos,])
predict(lm_fit,newdata=testing)
}
predictions<-rowMeans(predictions)
error<-sqrt((sum((testing$y-
predictions)^2))/nrow(testing))
```

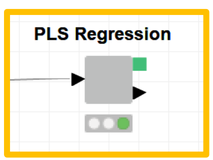
Coders:
Programs & Scripts

Data Science
Algorithm Inventors

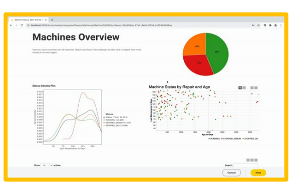


Data Scientists:
Visual Programming

Data Science
Creators



Citizen Data Scientists:
Components & Blueprints



Data Science Consumers:
Applications & Services

Data Science
Consumers

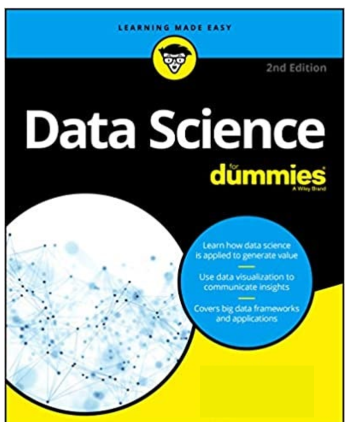
No Code, Low Code, Visual Programming?

No Code

No Code for Simple Problems:

No Code | (Code-Based) Programming

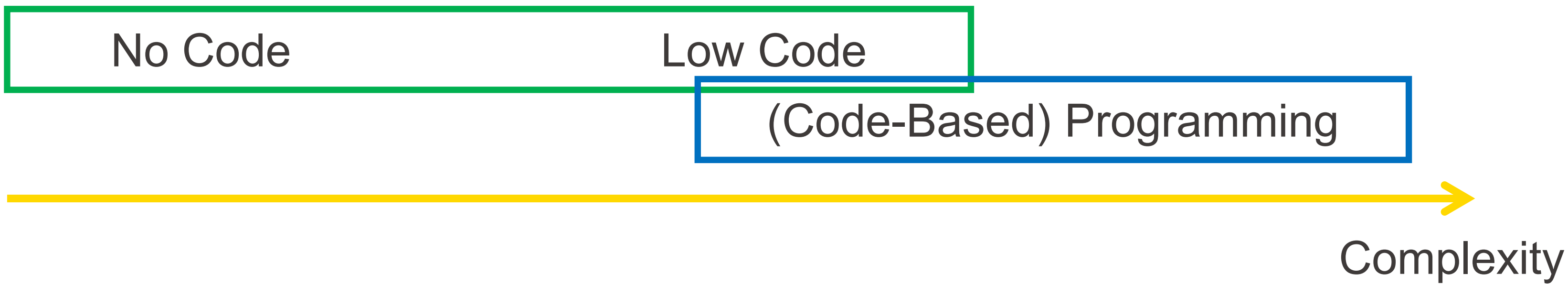
→ Complexity



Visual Environment allows quick creation of solutions for standard problems.

Low Code

Low Code for Standard Problems:

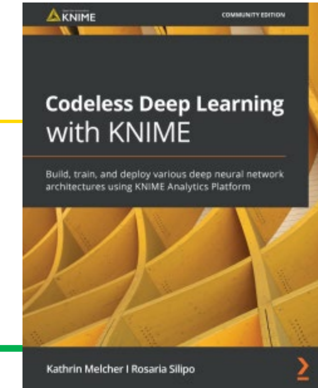


Visual Environment allows 80% creation of solutions.

…for the rest (& details): reach out to code.

(often: Visual UI on top of **one** programming language.)

Visual Programming



Visual Programming for all types of Problems:

Visual Programming

(Code-Based) Programming

Complexity

Visual Environment allows complete creation of solutions.
(from simple to complex)

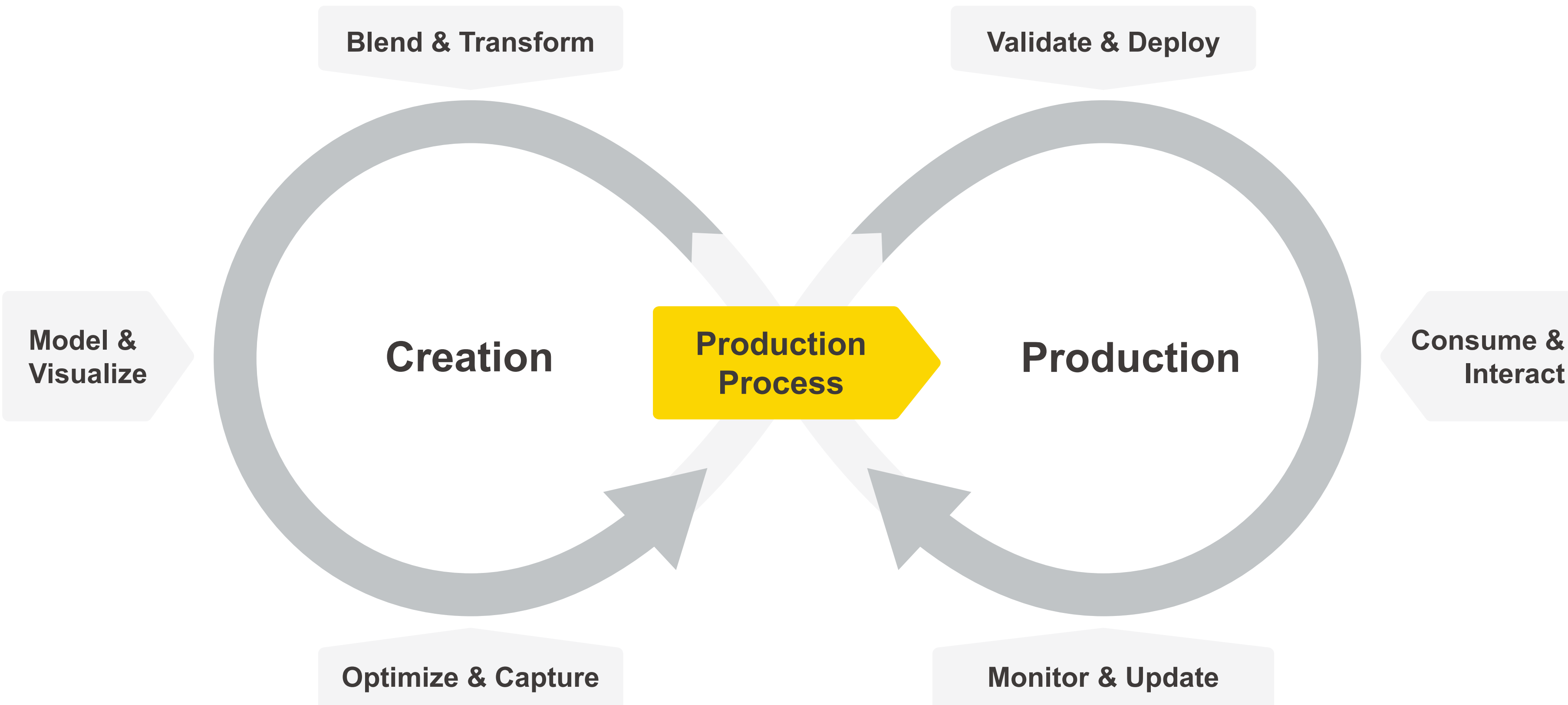
If desired: you can also include code (of various languages!).

(Visual UI models what's running under the hood.)

Open for Innovation KNIME

Visual Programming for Data Science

The Data Science Life Cycle



Data Science Activities

Data Wrangling
(Virtual Warehouses)

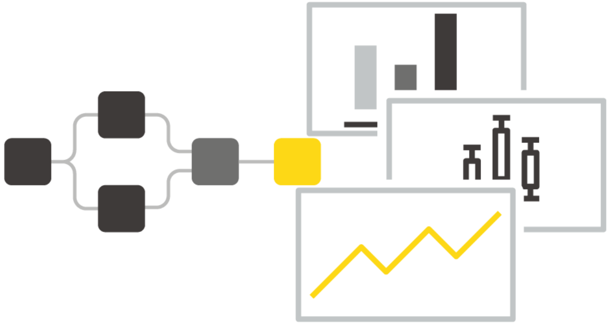
Descriptive Analytics
(Automating Repetition)

Diagnostic Analytics
(Creating Insights)

Advanced Analytics (AI/ML…)
(Predictive, Prescriptive)

Productionize
(Deploy, Manage, Govern)

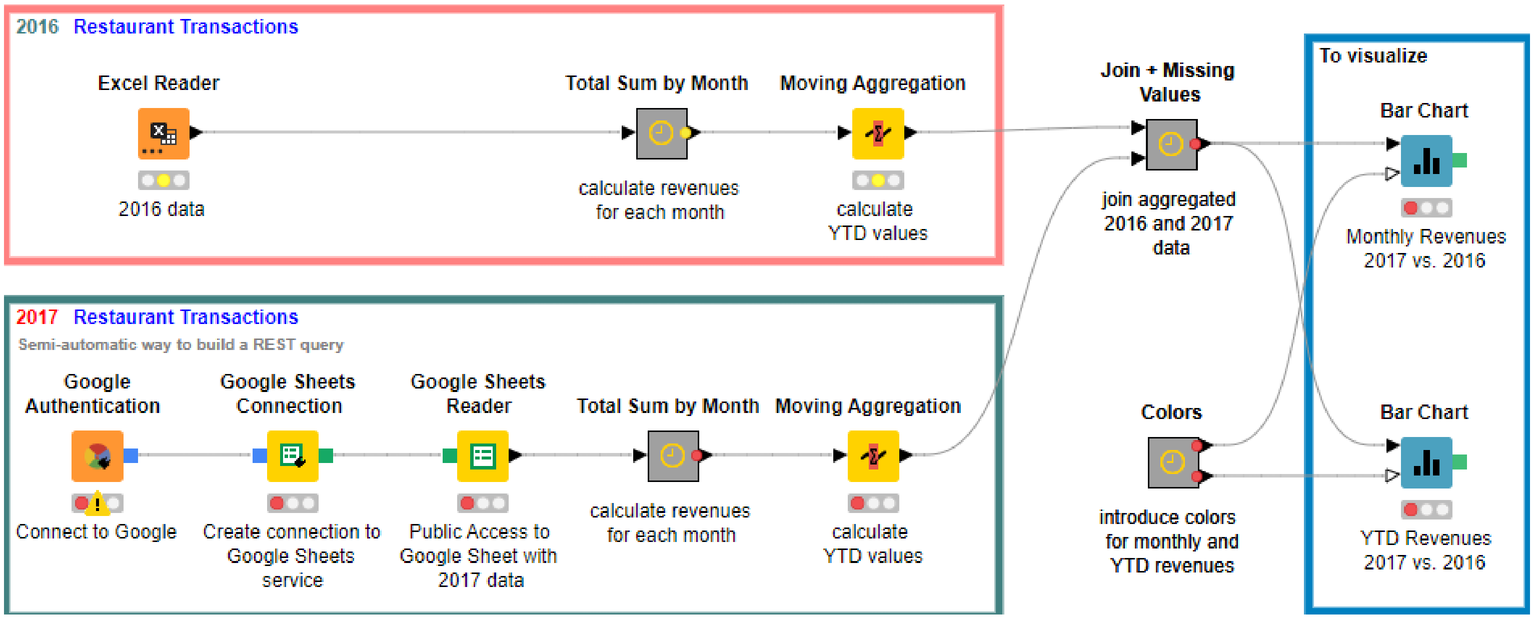
No Code for Analytics



No Code for Analytics

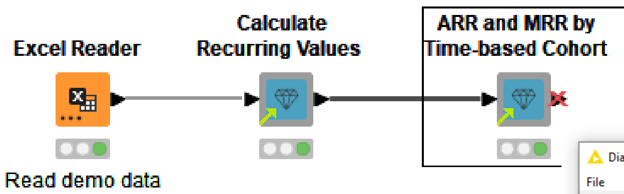
Benefits:

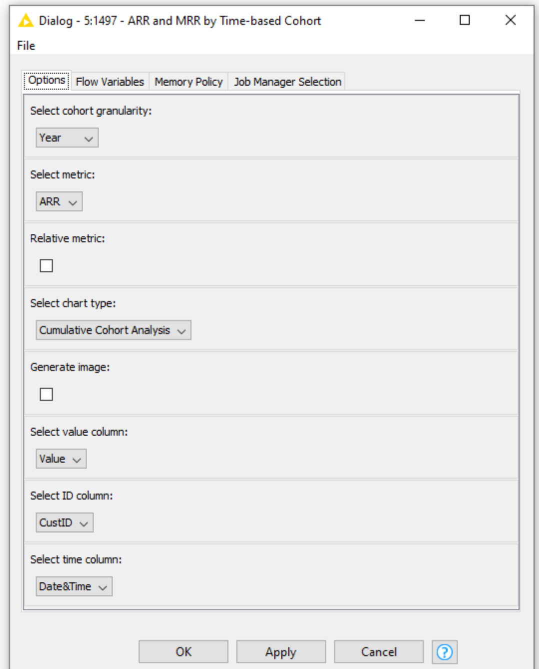
- (Reliable) Automation & Reproducibility
- Documentation

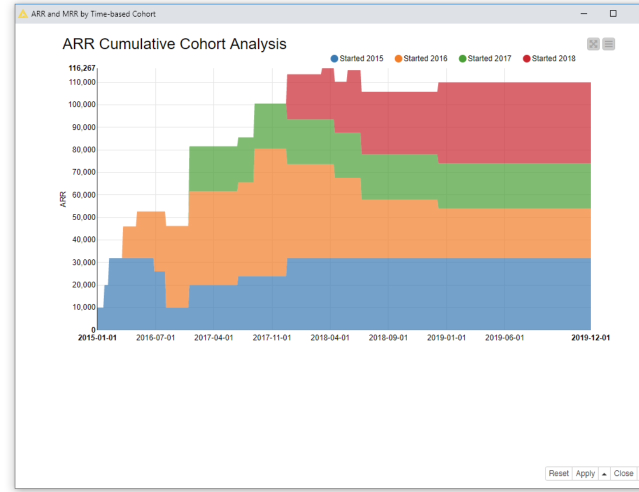


Creating Data Insights: Abstraction helps

ARR Cohort Analysis for Excel Users

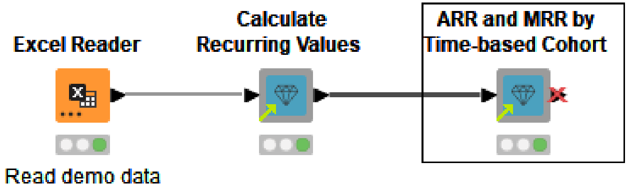


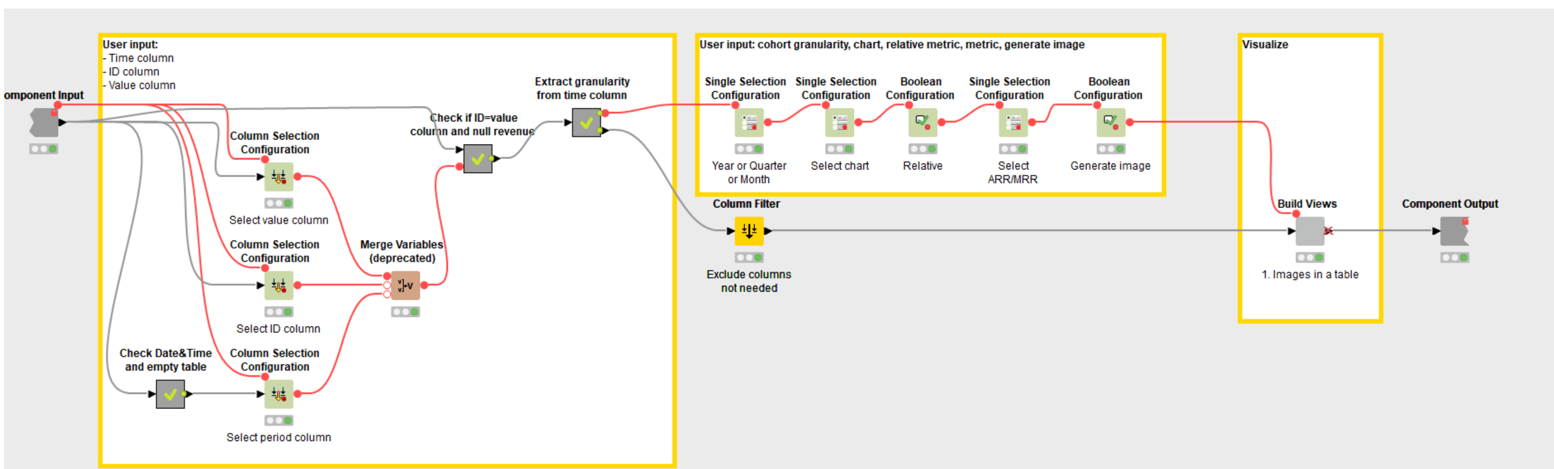




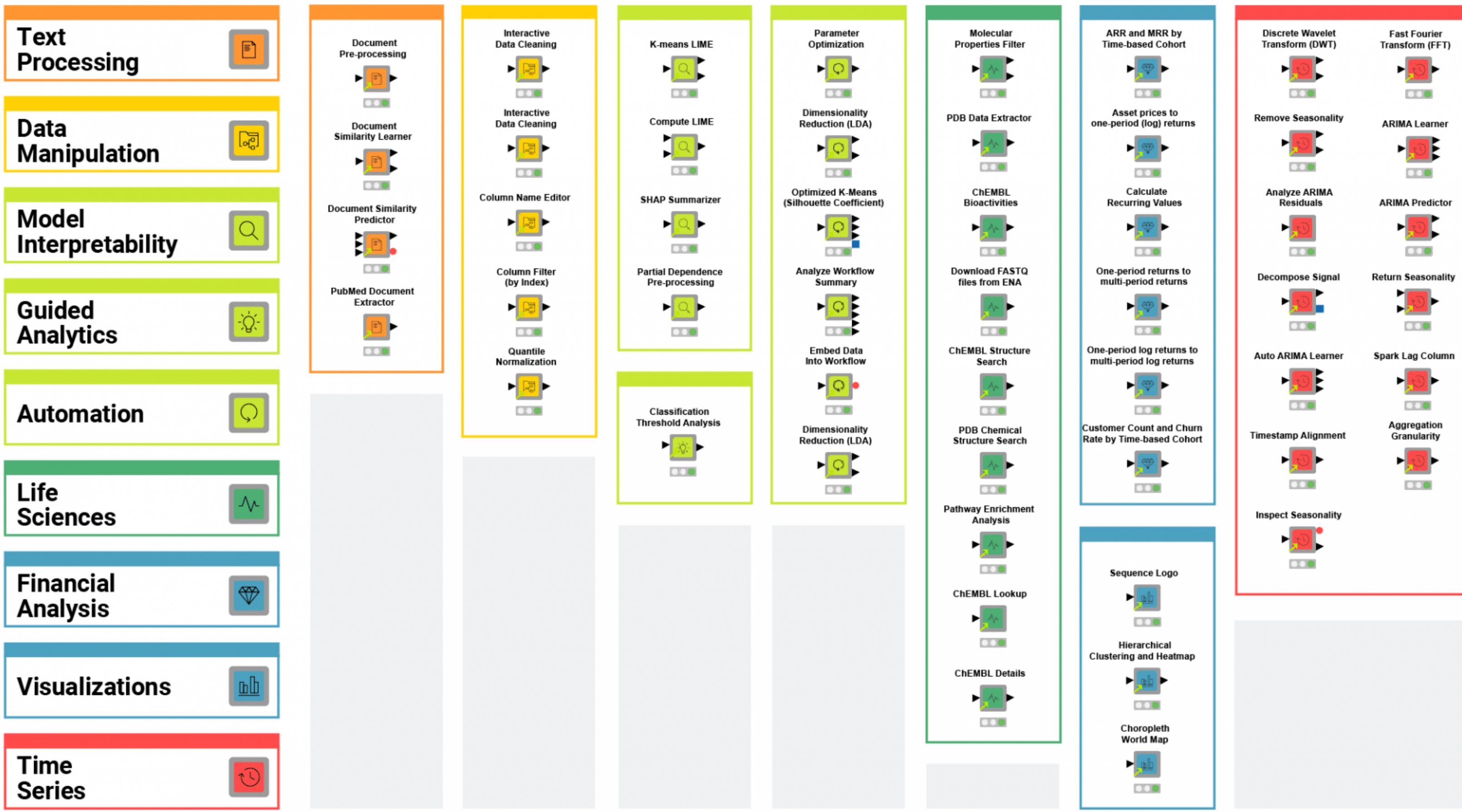
Creating Data Insights: Abstraction helps

ARR Cohort Analysis for Excel Users – No Code Encapsulated:



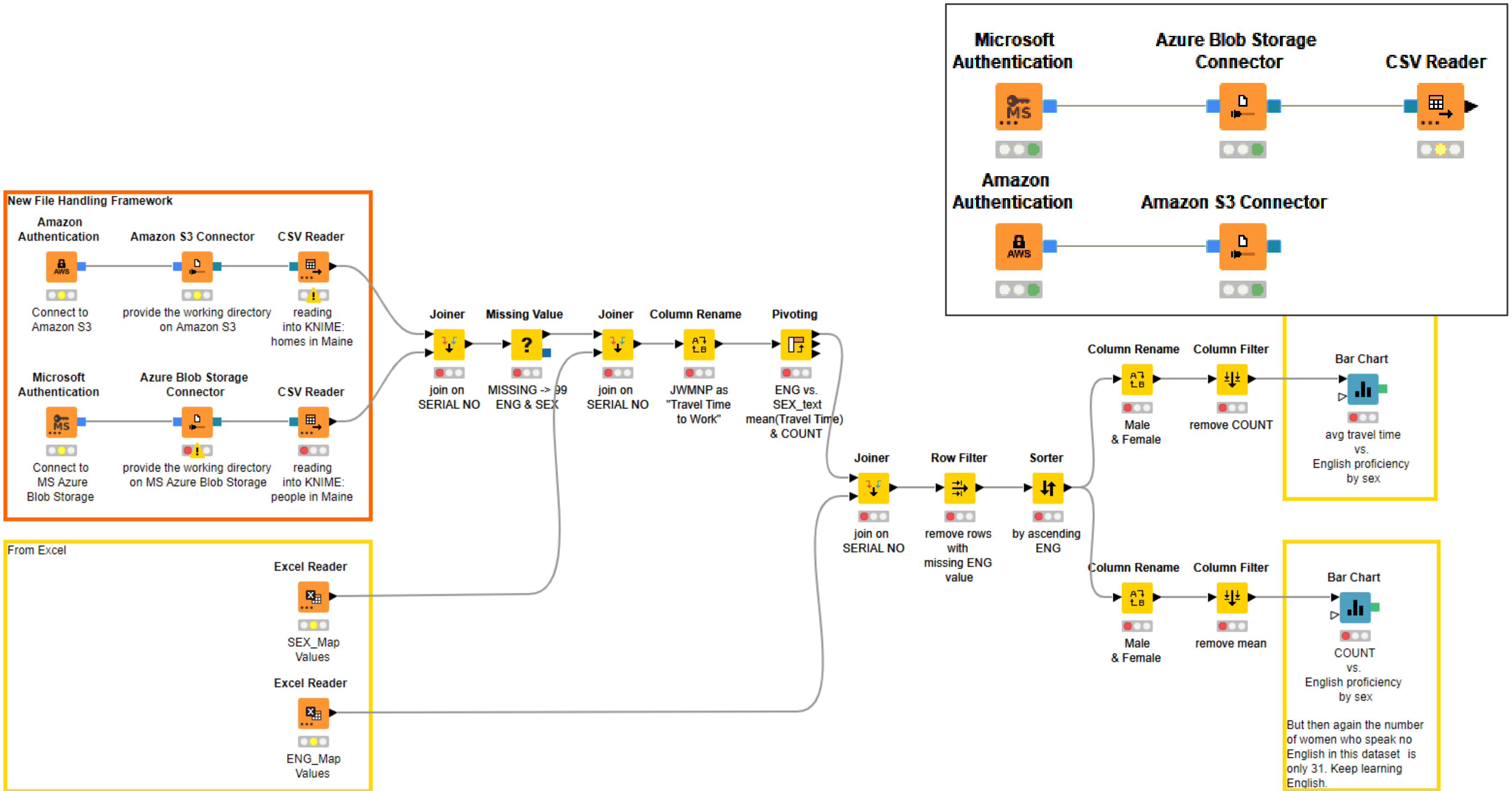


Abstraction in KNIME: Components

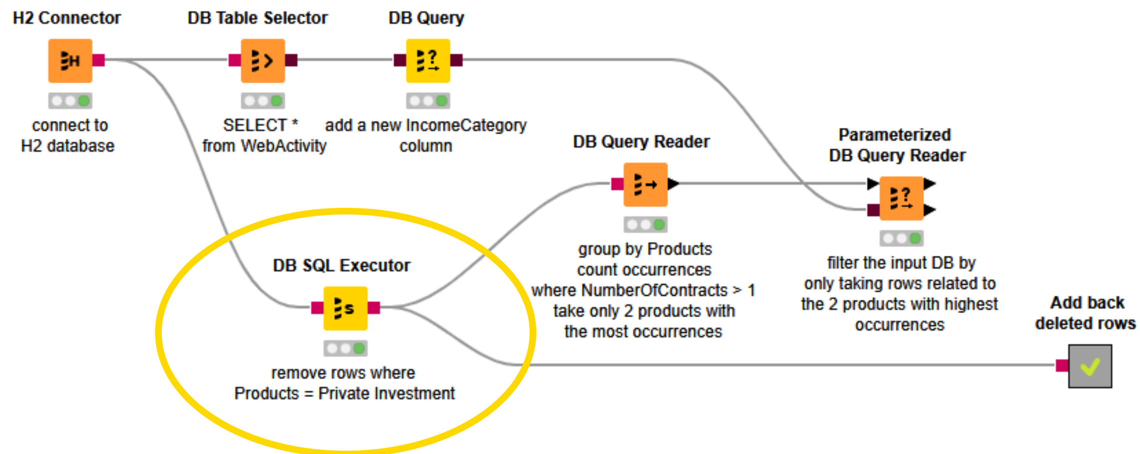


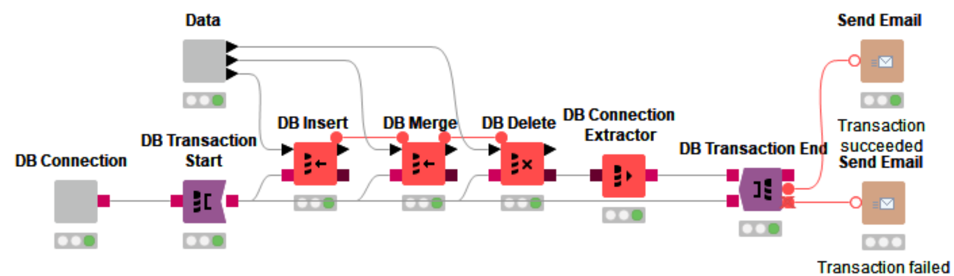
Low Code for Data Wrangling

Flexible Data Wrangling



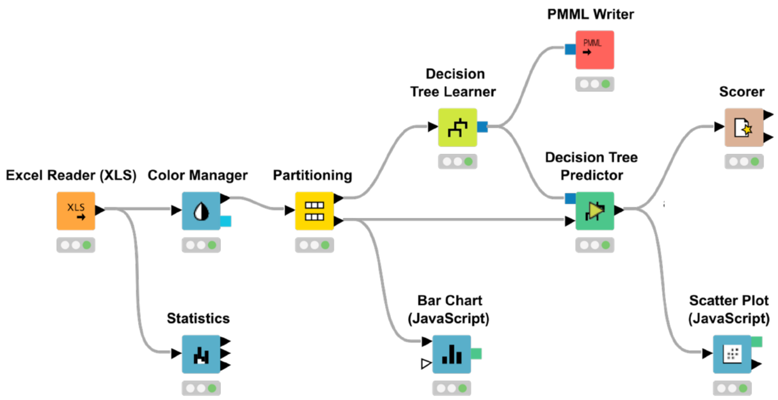
Flexible Data Wrangling

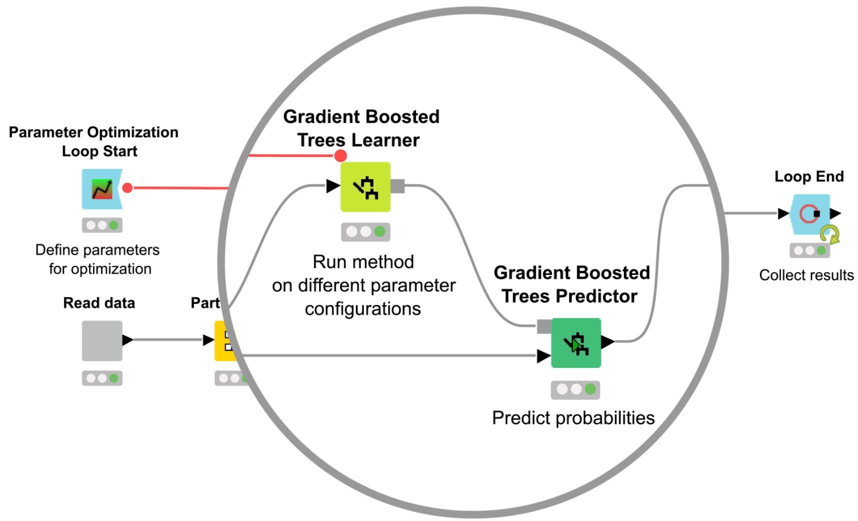




Low Code for Data Science

No Code for Data Science





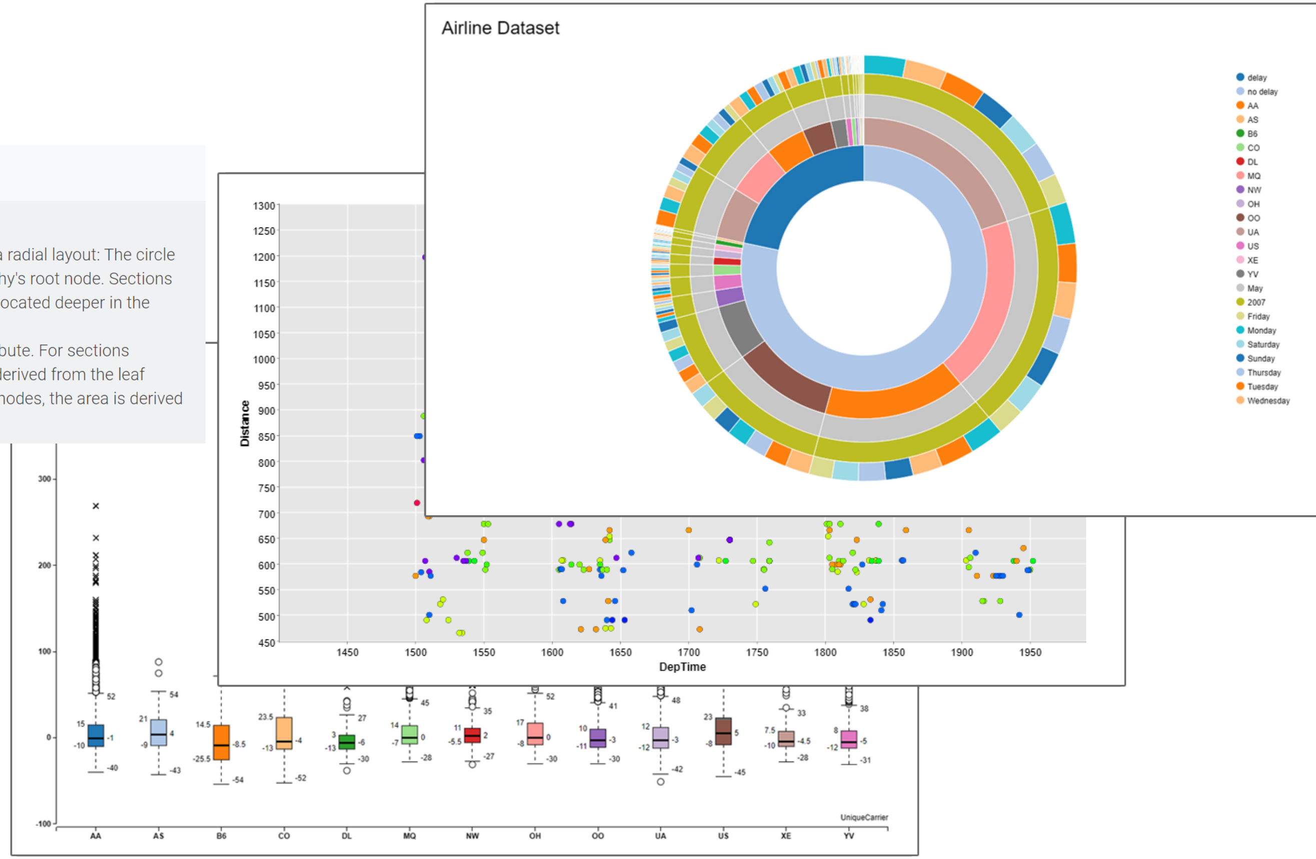
No Code for Interactive Visualizations

KNIME Hub › Nodes › Sunburst Chart

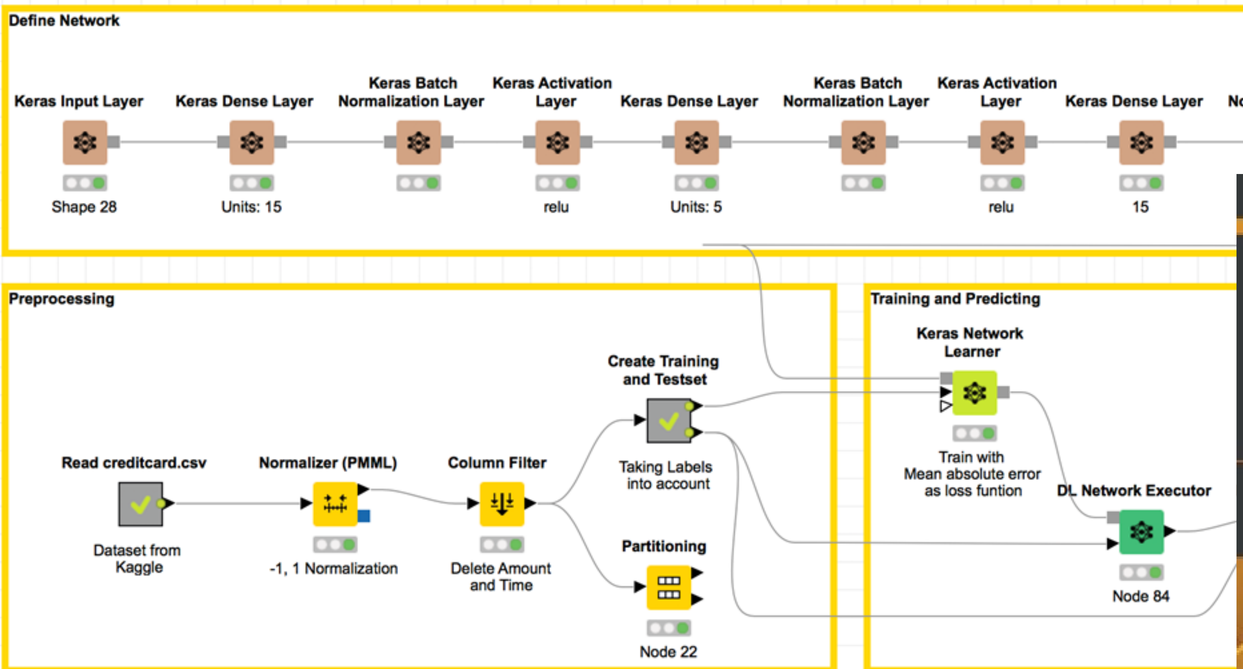
Node / Visualizer

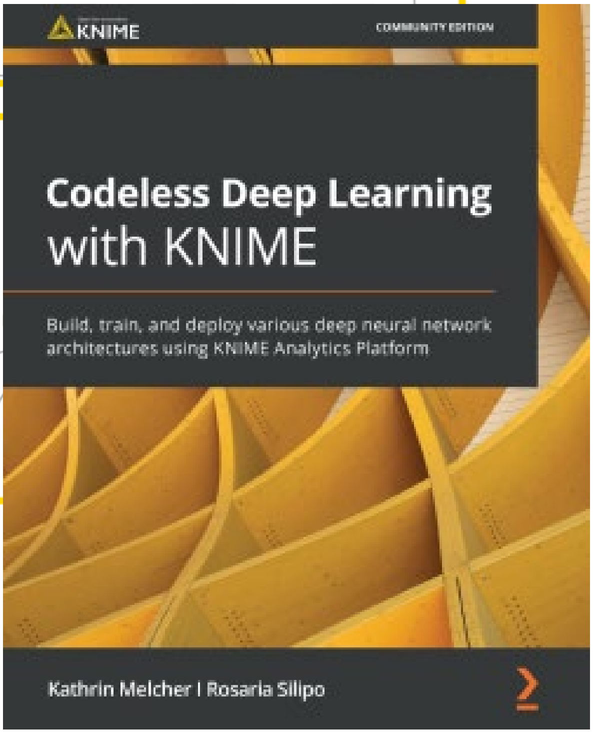
Sunburst Chart

This chart displays hierarchical data in a radial layout: The circle in the chart center represents the hierarchy's root node. Sections further outside represent nodes that are located deeper in the hierarchy.
Each leaf node has a attached value attribute. For sections corresponding to leaf nodes, the area is derived from the leaf node's value attribute. For sections not corresponding to leaf nodes, the area is derived from the accumulated value of all descending leaf nodes.



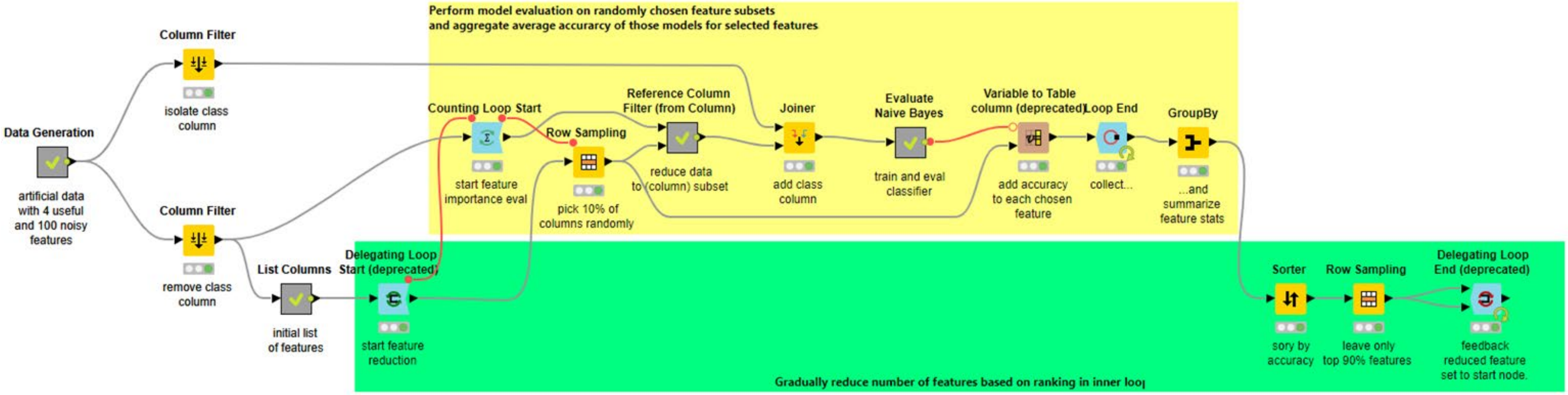
Codeless Deep Learning





Data Science *is* complex: Visual Programming

Visual Data Science Algorithms: Recursive Feature Elimination

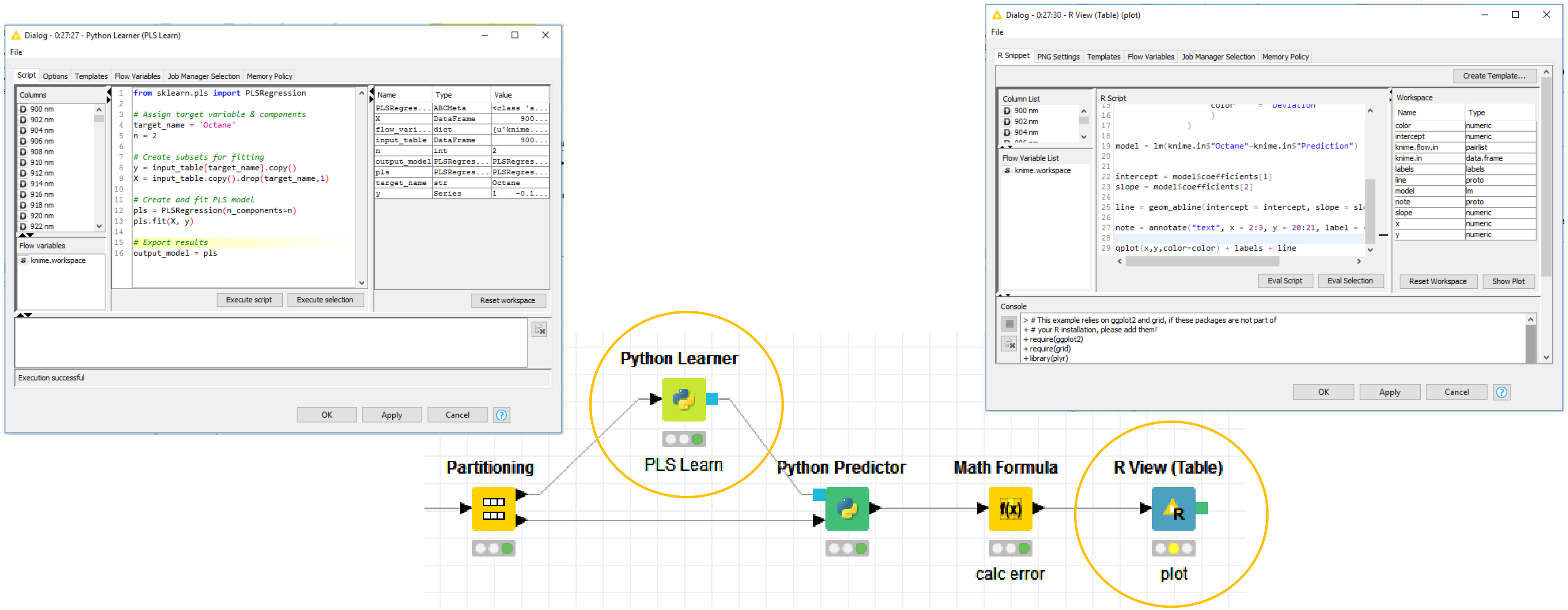


“Computer programming is an art, because it applies accumulated knowledge to the world, because it requires skill and ingenuity, and especially because it produces objects of beauty.”

Donald Knuth 1974

Low Code for Data Science

Leveraging Code in Visual Programming:



…works with (and from…) Jupyter Notebooks as well.

Data Science *is* complex: More Topics

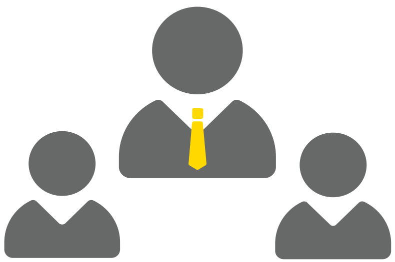
Data Science "Stuff":

- Data Blending:
  - Feature Selection
  - Feature Engineering
  - Anonymization
  - …
- Modeling
  - Model Selection
  - Parameter Optimization
  - Ensemble Creation
  - Active Learning
  - Transfer Learning
  - …

Management Requirements:

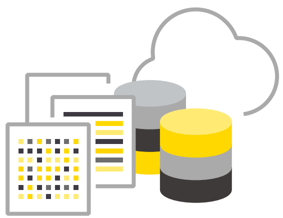
- Compliance
  - Best Practices
  - Required Standards
  - Bias Detection and Removal
  - Data Privacy / GDPR
  - Explainable "AI"
  - …
- Governance
  - Integration / Dependencies
  - Traceable / Lineage
  - Documented
  - Reproducibility
  - …

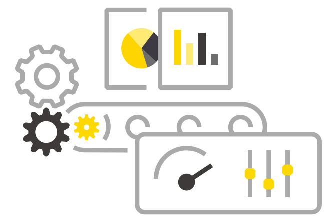
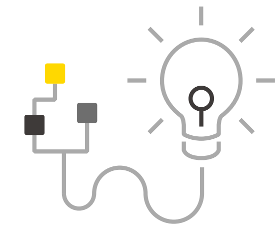
Data Science Teams



The Data Science Life Cycle

Blend & Transform	Model & Visualize	Deploy & Manage	Consume & Interact



Techniques:

- Joins
- Pivot
- Normalization
- PCA
- …

- Clustering
- Regression
- Deep Learning
- Visualizations
- …

- Model Monitoring
- Updating
- Validation
- …

- Active Learning
- Interactive Applications
- Services
- …

Technologies and Languages:

- SQL
- Data Lakes (AWS, Azure, Google…)
- …

- R, Python
- Java, C
- JavaScript, …

- Go
- Kubernetes
- Docker
- Grafana
- …

- Plotly, R-Shiny
- JavaScript
- Vue.js
- REST
- …

Data Science Teams shouldn't really have to worry about all of this…

Democratizing Data Science

What's the Essence of the Job:

- Create (complex) Data Science processes, often together with other Experts

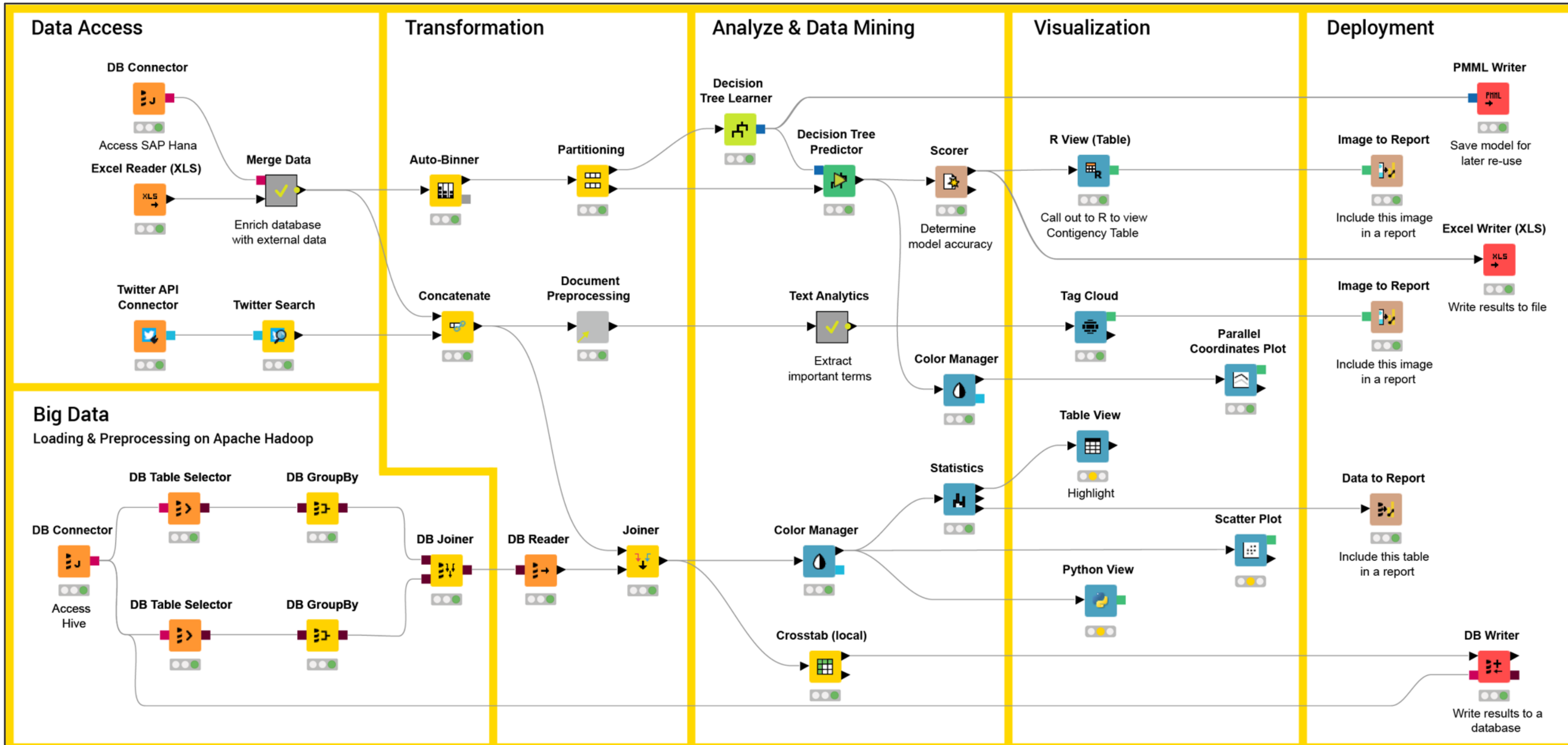
What's usually not part of the Job:

- Inventing, Writing, Optimizing new Algorithms

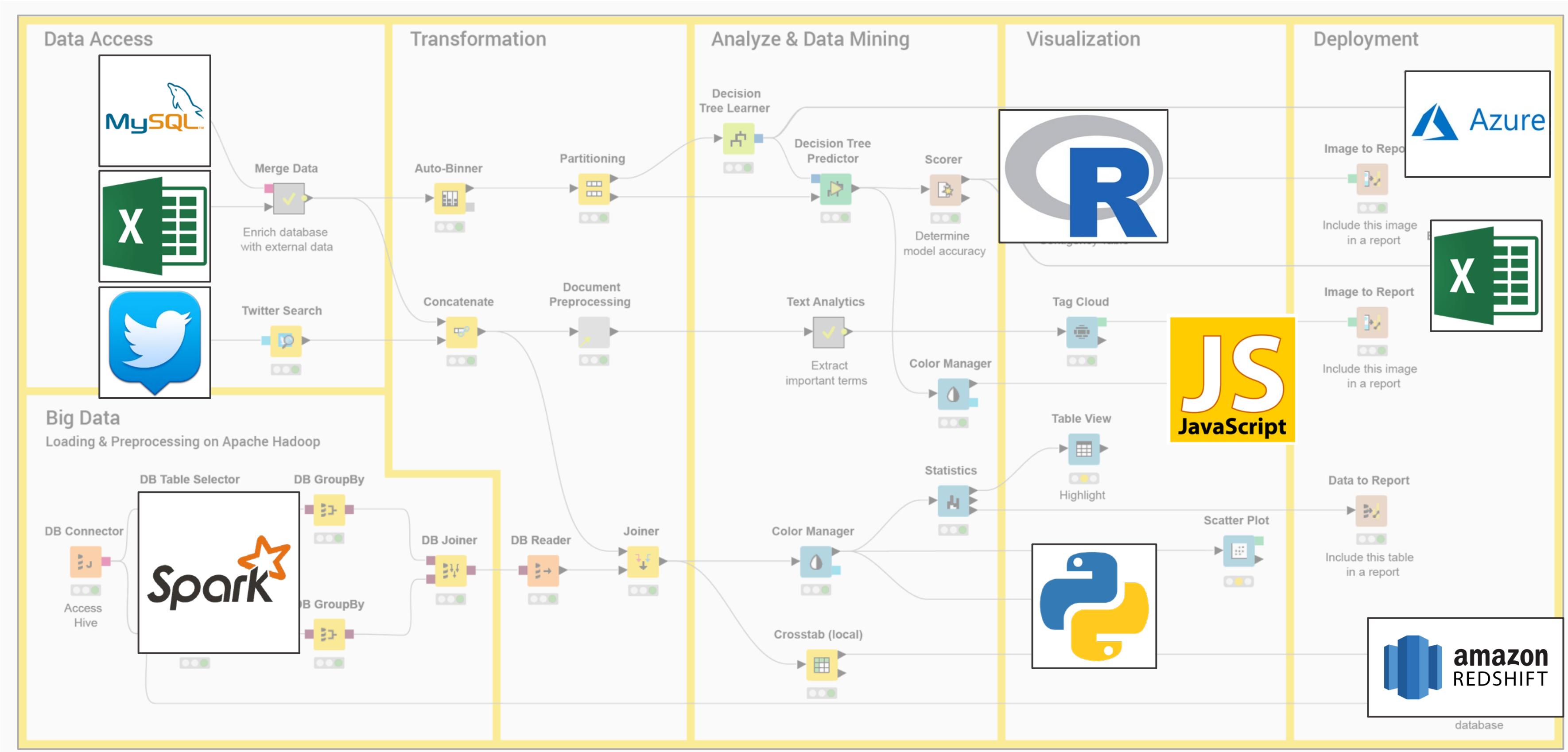
What shouldn't be part of the Job:

- Caring about the details of the underlying technology
- Worrying about interfaces of different tools
- Worrying about library versions and (backwards) compatibility
- Manually doing repetitive tasks

Visual Programming for Data Science

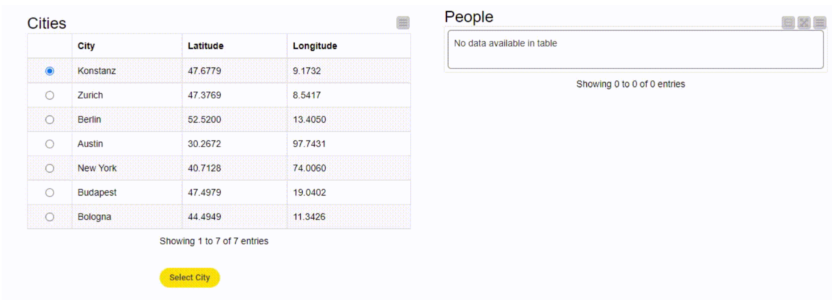


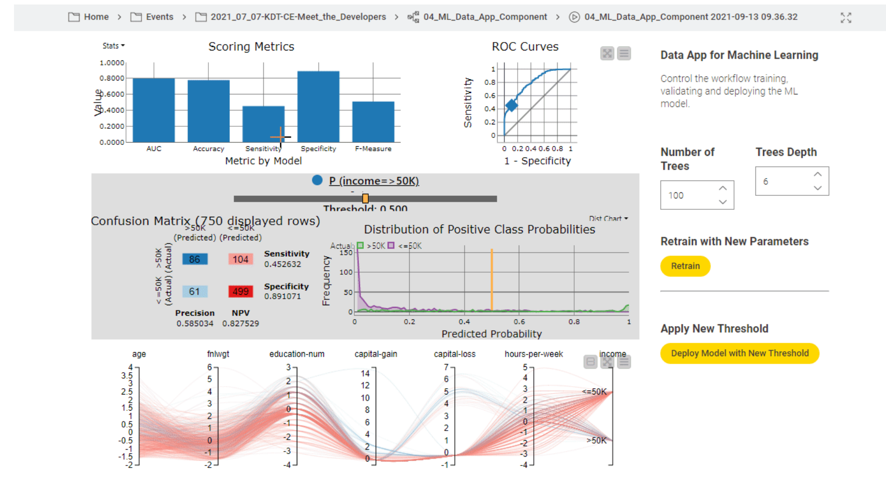
Technologies & Languages under the Hood

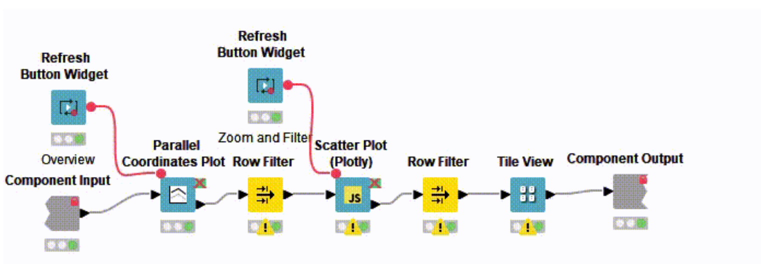


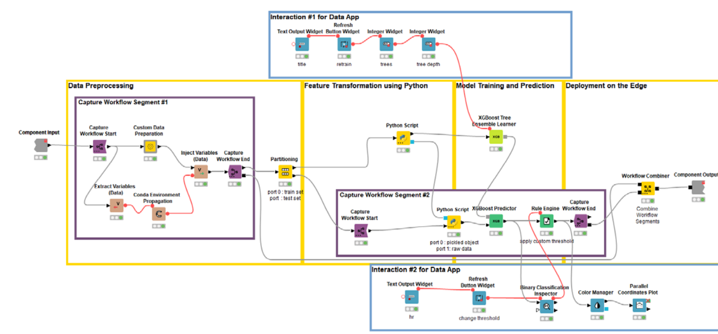
Low Code for Deployment

No Code Deployment of Data Apps



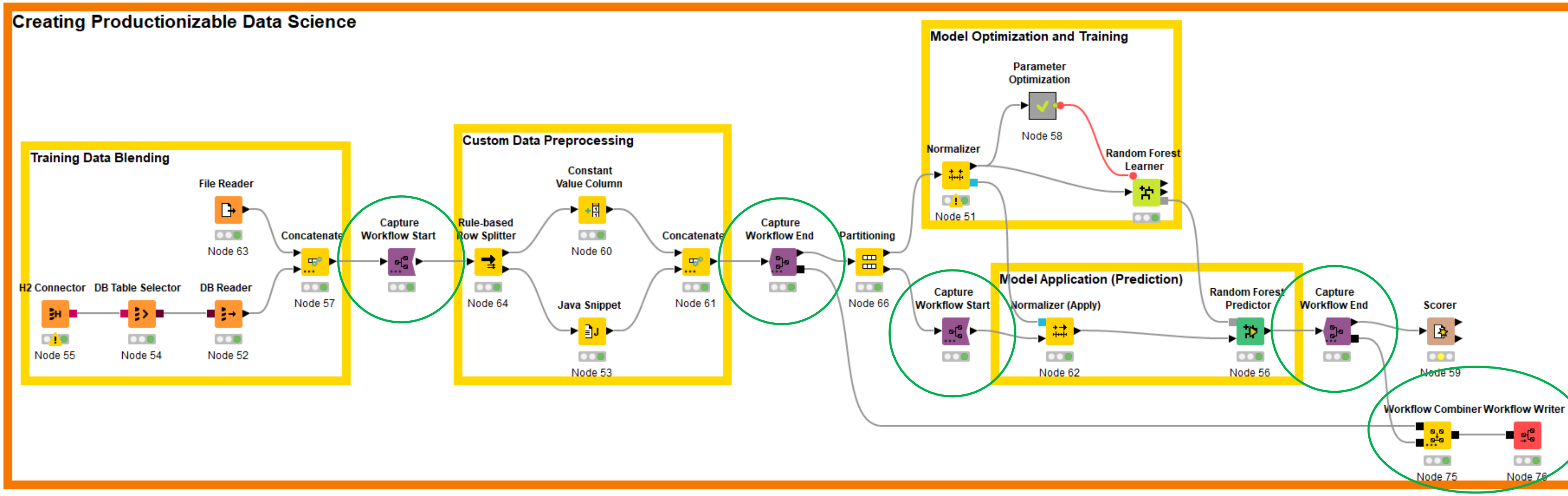




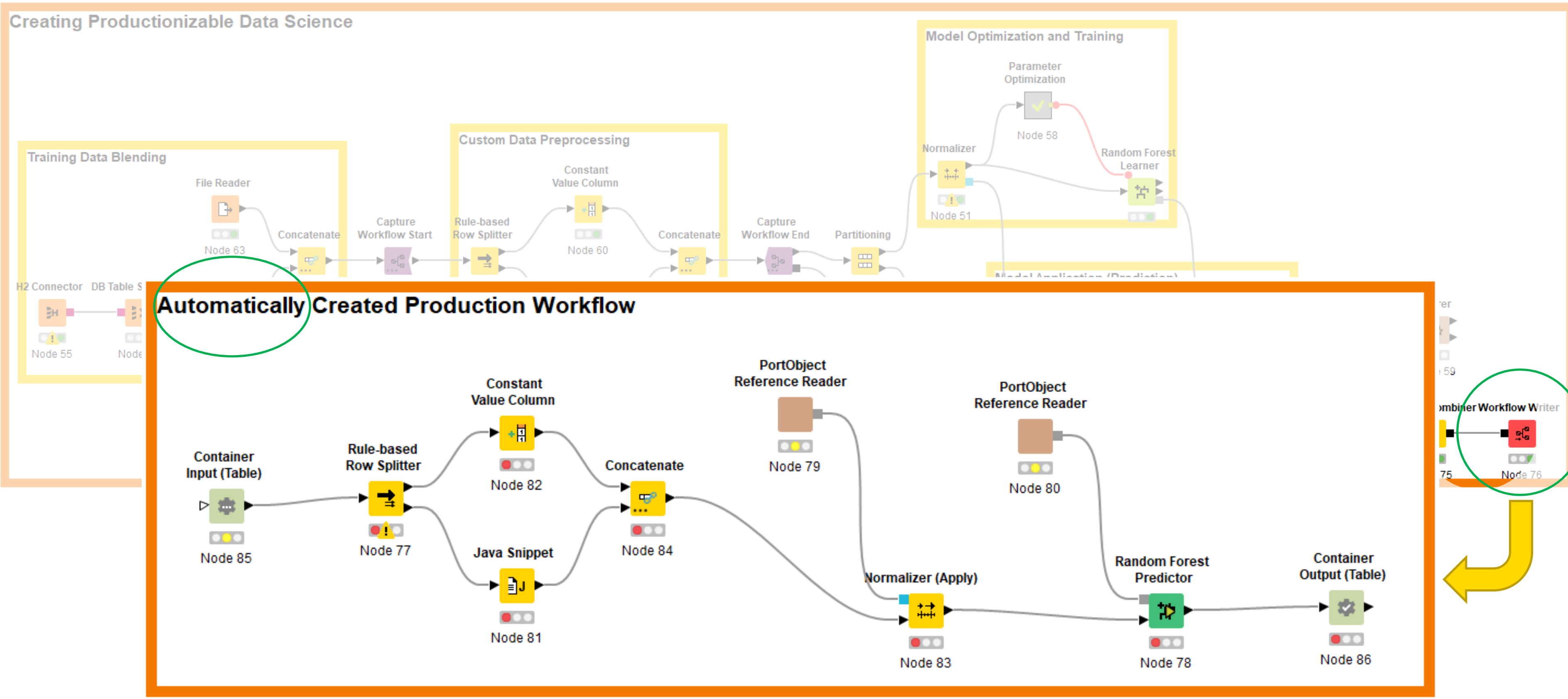


Open for Innovation KNIME

Deploying Data Science as Service



Deploying Data Science as Service



Visual Programming for all Aspects of Data Science

Data Wrangling
(Virtual Warehouses)

Descriptive Analytics
(Automating Repetition)

Diagnostic Analytics
(Creating Insights)

Advanced Analytics (AI/ML…)
(Predictive, Prescriptive)

Productionize
(Deploy, Manage, Govern)

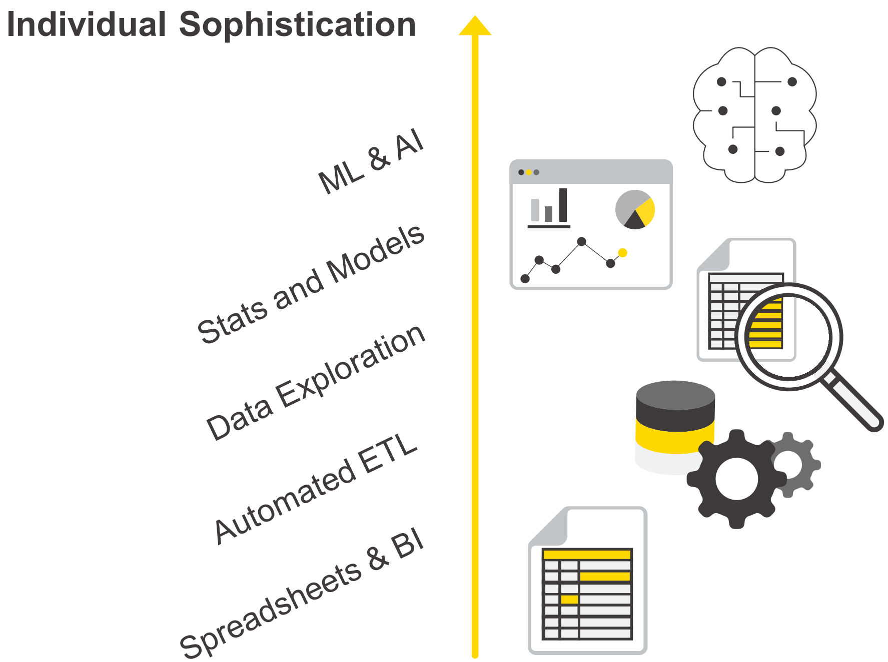
Visual Programming for Data Science makes Sense!

- Data Science is a Team Sport:
  - requires a broad spectra of skills – often done in collaboration
  - requires understanding and access to the inner wheels of an algorithm (but not to the algorithm itself)
  - No single language or tool has all the answers – it's about tool blending as well!

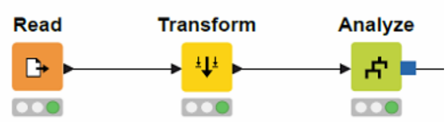
- No-Code, Low-Code, Visual Programming (if done right):
  - automates repetitive tasks
  - provides the appropriate level of abstraction to allow focus on modeling a data process
  - allows access to all relevant nuts and bolts
  - allows to abstract / encapsulate sophistication for others to safely reuse
  - removes tool interaction/interface complexity
  - provides transparency & governance
  - (and it's not just a UI slapped on top of a language)

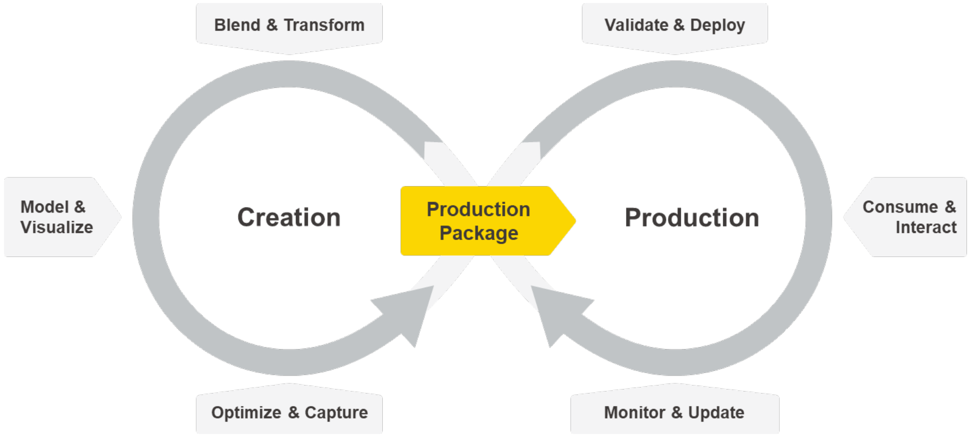
  - …allows everybody to gradually increase their Data Science Proficiency (in the same Environment).

Data Science Upskilling



Organizational Upskilling





Organizational Maturity

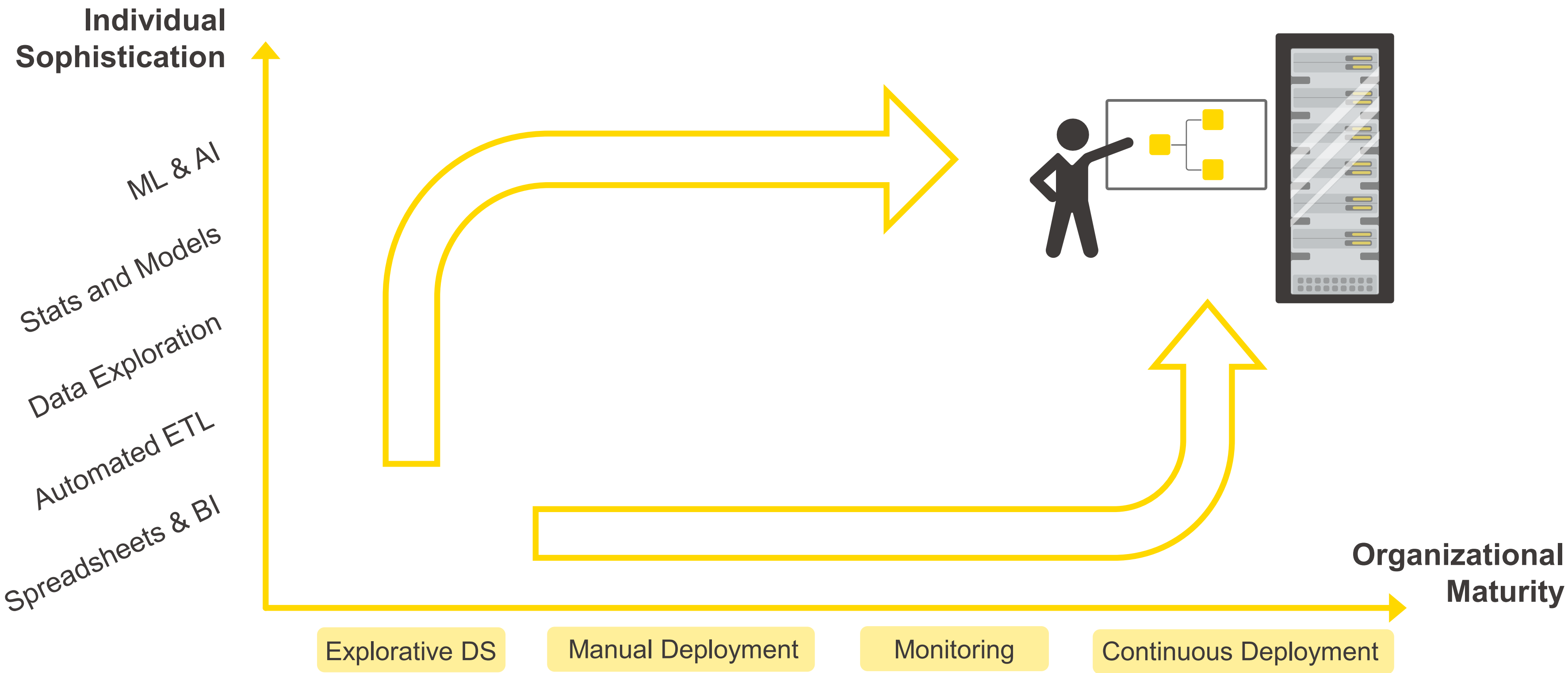
Explorative DS

Manual Deployment

Monitoring

Continuous Deployment

Upskilling Individuals and Organizations



Want to get started?

Getting Started?